EPiC Series in Computing
Volume 63, 2019, Pages 101–110
Proceedings of 32nd International Conference on Computer Applications in Industry and Engineering

Stock Price Prediction Using LSTM on Indian Share Market

Achyut Ghosh[1], Soumik Bose[1], Giridhar Maji[2], Narayan C. Debnath[3], Soumya Sen[1]
[1]A.K. Chowdhury School of I.T., University of Calcutta
[2]Asansol Polytechnic, Asansol, India
[3]Department of Software Engineering, Eastern International University, Vietnam
achyutghosh06@gmail.com,1994bolusoumik@gmail.com, giridhar.maji@gmail.com, narayan.debnath@eiu.edu.vn, iamsoumyasen@gmail.com

Abstract

Predicting stock market is one of the most difficult tasks in the field of computation. There are many factors involved in the prediction – physical factors vs. physiological, rational and irrational behavior, investor sentiment, market rumors,etc. All these aspects combine to make stock prices volatile and very difficult to predict with a high degree of accuracy. We investigate data analysis as a game changer in this domain.As per efficient market theory when all information related to a company and stock market events are instantly available to all stakeholders/market investors, then the effects of those events already embed themselves in the stock price. So, it is said that only the historical spot price carries the impact of all other market events and can be employed to predict its future movement. Hence, considering the past stock price as the final manifestation of all impacting factors we employ Machine Learning (ML) techniques on historical stock price data to infer future trend. ML techniques have the potential to unearth patterns and insights we didn't see before, and these can be used to make unerringly accurate predictions. We propose a framework using LSTM (Long Short-Term Memory) model and companies' net growth calculation algorithm to analyze as well as prediction of future growth of a company.

1 Introduction

Data analysis have been used in all business for data-driven decision making. In share market, there are many factors that drive the share price, and the pattern of the change of price is not regular. This is why it is tough to take a robust decision on future price. Artificial Neural Network (ANN) has the capability to learn from the past data and make the decision over future. Deep learning networks

Q. Yuan, Y. Shi, L. Miller, G. Lee, G. Hu and T. Goto (eds.), CAINE 2019 (EPiC Series in Computing, vol. 63), pp. 101–110

such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) etc. works great with multivariate time series data. We train our model from the past stock data and calculate the future price of that stock. This future price use to calculate the future growth of a company. Moreover, we found a future growth curve from different companies. Thus we can analyze and investigate the similarity of one company's future curve over another. Stock price of a listed company in a stock exchange varies every time an order is placed for sell or buy and a transaction completes. An exchange collects all sell bids with expected price per stock (normally it is more than the price paid while bought by the investor) and all buy bids with or without a price limit (normally an investor expects the future price of the stock will be more than the current price he is paying now) and a buy-sell transaction is committed when both bids have a match i.e. selling bid price is same with buying bid price of some buy-bid Fama in 1970 [1] proposed efficient market hypothesis which says that in an efficient market (where all events are known to all stakeholders as an when it happens) the effect all market events are already incorporated in stock prices hence it is not possible to predict using past events or prices. The stock price of a company depends on many intrinsic as well as extrinsic attributes. Macro-economic conditions too play an important role in growth or decline of a sector as a whole. Some of the intrinsic factors could be company's net profit, liabilities, demand stability, competion in market, technically advanced assembly line, surplus cash for adverse situations, stakes in raw material supplier and finished product distributors etc. Those factors that are beyond the control of the company such as crude oil price, dollar exchange rate, political stability, government policy decision etc. come under extrinsic attribute. Many researchers have tried using the historical stock prices as the basis for time series analysis to forecast future stock prices. Many different statistical models were applied since long like moving average (MA), autoregression (AR), weighted moving average, ARIMA, CARIMA etc. Later some non linear models were also tried like GARCH. Recently different neural network models, evolutionary algorithms wre being applied for stock prediction with success. Deep neural networks like CNN, RNN are also used with different parsmeter settings and features. In this paper we shall explore a special type of RNN known as LSTM to predict future company growth based on past stock prices.

2 Indian Stock Market Overview

Almost every country has one or more stock exchanges, where the shares of listed companies can be sold or bought. It is a secondary market place. When a company first lists itself in any stock exchange to become a public company, the promoter group sells substantial amount of shares to public as per government norms. During incorporation of a company shares are bought by promoter groups or institutional investors in a primary market. Once promoter offloads major portion of the shares to public relail investors, then those could be traded in secondary market i.e. in stock exchanges. In India the BSE(Bombay Stock Exchange) and the NSE(National Stock Exchange) are the two most active stock exchange. The BSE has around 5000 listed companies where as NSE had around 1600. Both the exchange has similar trading mechanism and market open time, closing time and settlement process. Stock exchanges helps individual investors to take part in the share market and allows to buy even a single share of some listed company with the help of a trading account and demat account. These online markets have revolutionized the Indian investment arena along with government initiative like tax benefit on equity investment, National Pension Scheme (NPS) investing in share market etc. Due to continuous reduction in bank inrest rates and increasing inflation middle class investors are moving towards equity market from the safe heaven of fixed deposites. All these have helped to grow the capitalization of both the exchanges.

3 Related Studies

There are lots of research work in stock market prediction as well as in LSTM. Almost every data mining and prediction techniques were applied for prediction of stock prices. Many different features and attributes were used for the same purpose. There are three main categories of stock market analysis and prediction such as (a) Fundamental analysis, (b) Technical analysis and (c) Time series analysis. Most of the stock forecasting techniques with time series data normally use either a linear such as AR, MA, ARIMA, ARMA, CARIMA, etc. [1],[2] or non-linear models (ARCH, GARCH, ANN, RNN, LSTM, etc.).Authors in [3] have analyzed many different macro-economic factors by designing a data warehouse that affects share price movement such as crude oil price, exchange rate, gold price, bank interest rate, political stability, etc. Researchers in [4] employed frequent itemset mining technique to find a lagged correlation between price movement between different sectorial index in Indian share market. Roondiwala et al. in [5] has used RNN-LSTM model on NIFTY-50 stocks with 4 features (high/close/open/low price of each day). They have used 21 days window to predict the next day price movement. A total of 5 years data has been used for prediction and RMSE as error metric to minimize with backpropagation.

Kim et al. in [6] proposed a model, 'the feature fusion long short-term memory-convolutional neural network (LSTM-CNN) model'. They have used CNN to learn the features from stock chart images. They found that the candlestick charts are the best candidate for predicting future stock price movement. Next they employed LSTM and fed with historical price data. They have tested on minute-wise stock price and used 30 minute sliding window to forecast 35^{th} minute price. They have tested on S&P 500 ETF data with stock price and trade volume using CNN. They use the CNN and LSTM individually on different representation of the same data and then used the combined feature fusion model for the same purpose. It is observed that the combined model outperforms individual models with less prediction error. Thus this work establishes the fact that different representation of the same data (raw stock price and trade volume and stock chart image) with combined models where each individual model is optimized for separate data format can learn more intrinsic data dynamics and features which is alalogous to looking on the same object from different perspective angles that gives new insight. Hiransha et al.in their paper [7], employed three different deep learning network achitectures such as RNN, CNN and LSTM to forecast stock price using day wise past closing prices. They have considered two company from IT sector (TCS and Infosys) and one from Pharma sector(Cipla) for experiment. The uniqueness of the study is that they have trained the models using data from a single company and used those models to predict future prices of five different stocks from NSE and NYSE (Newyork Stock Exchange). They argued that linear models try to fit the data to the model but in deep networks underlying dynamic of the stock prices are unearthed. As per their results CNN outperformed all other models as well as classical linear models. The DNN could forecast NYSE listed companies even though the model has learned from NSE dataset. The reason could be the similar inner dynamics of both the stock exchanges. Gers & Schmidhuber proposed a variation of LSTM by introducing "peephole connections" [18]. In this model the gate layers can look into the cell state. In another case the model coupled forget and input gates. In this case, decision to add new information or to forget it is taken together. It forgets only when it needs to input something in its place. This architecture inputs new values to the cell state when it forgets anything older. Cho, et al. [19] proposed another popular LSTM variation known as the Gated Recurrent Unit (GRU). It aggregates both the forget and input gates into an "update gate." The cell state and hidden state are merged along with a few other minor modifications to make the final model more simple than the original LSTM. Due to the above reason this model is becoming popular day by day. These are by no means an exhaustive list of modified-LSTMs. There are many other variants such as Depth Gated LSTMs by Yao, et al. [20]. Koutnik, et al. [21] proposed 'Clockwork RNNs' to tackle long-term dependencies in a completely different manner.

4 LSTM Architecture

4.1 An overview of Recurrent Neural Network (RNN)

In a classical neural network, final outputs seldom act as an output for the next step but if we pay attention to a real-world phenomenon, we observe that in many situations our final output depends not only the external inputs but also on earlier output. For example, when humans read a book, understanding of each sentence depends not only on the current list of words but also on the understanding of the previous sentence or on the context that is created using past sentences. Humans don't start their thinking from scratch every second. As you read this essay, you understand each word based on your understanding of previous words. This concept of 'context' or 'persistence' is not available with classical neural networks. Inability to use context-based reasoning becomes a major limitation of traditional neural network. Recurrent neural networks (RNN) are conceptualized to alleviate this limitation.RNN are networked with feedback loops within to allow persistence of information. The Figure 1**Error! Reference source not found.** shows a simple RNN with a feedback loop and its unrolled equivalent version side by side.

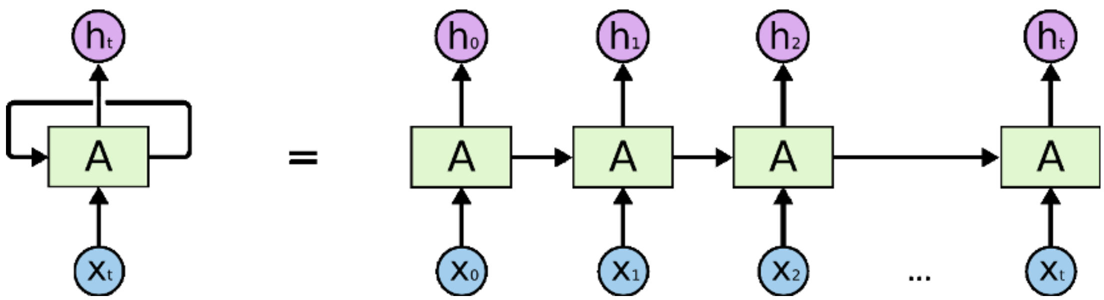

Figure 1: An unrolled recurrent neural network

Initially (at time step t) for some input X_t the RNN generates an output of h_t. In the next time step (t+1) the RNN takes two input X_{t+1} and h_t to generate the output h_{t+1}. A loop allows information to be passed from one step of the network to the next.

RNNs are not free from limitations though. When the 'context' is from near past it works great towards the correct output. But when an RNN has to depend on a distant 'context' (i.e. something learned long past) to produce correct output, it fails miserably. This limitation of the RNNs was discussed in great detail by Hochreiter [8] and Bengio, et al. [9]. They also traced back to the fundamental aspects to understand why RNNs may not work in long-term scenarios. The good news is that the LSTMs are designed to overcome the above problem.

4.2 LSTM Networks

Hochreiter & Schmidhuber [10] introduced a special type of RNN which is capable of learning long term dependencies. Later on many other researchers improved upon this pioneering work in [11] [12] [13] [14]. LSTMs are perfected over the time to mitigate the long-term dependency issue. The evolution and development of LSTM from RNNs areexplained in [15] [16].

Recurrent neural networks are in the form of a chain of repeating modules of the neural network. In standard RNNs, this repeating module has a simple structure like a single *tanh* layer as shown in Figure 2.

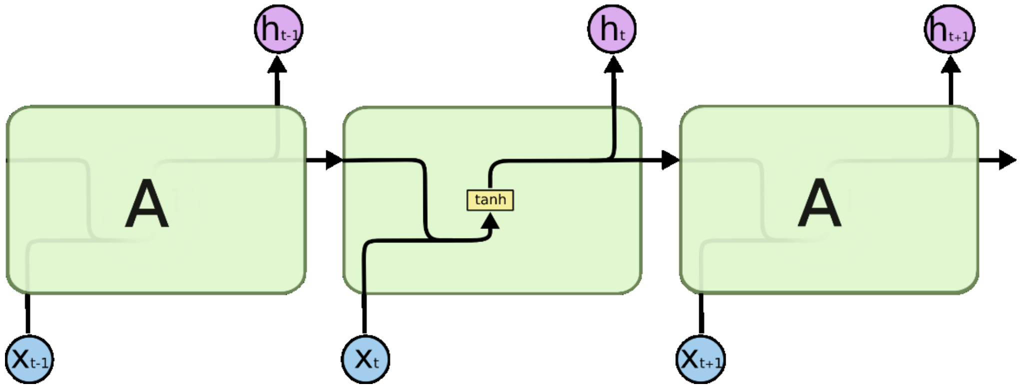

Figure 2: The repeating module in a standard RNN contains a single layer

LSTMs follow this chain-like structure, however the repeating module has a different structure. Instead of having a single neural network layer, there are four layers, interacting in a very special way as shown in Figure 3.

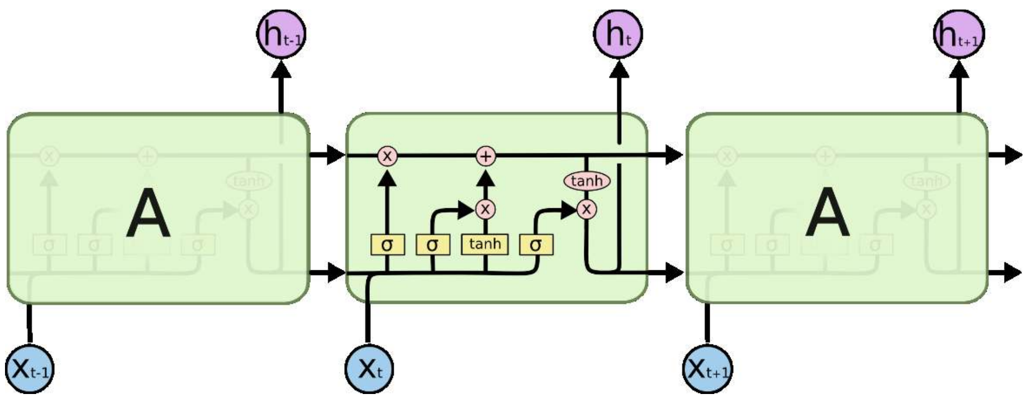

Figure 3: The repeating module in an LSTM contains four interacting layers

In Figure 3, every line represents an entire feature vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denotes its content being copied and the copies going to different locations.

4.3 The Working of LSTM

The key to LSTMs is the cell state, the horizontal line running through the top of the diagram.The cell state is like a conveyor belt. This runs straight down the entire chain, having some minor linear interactions. LSTM has the ability to add or remove information to the cell state, controlled by structures called gates. Gates are used for optionally let information through. Gates are composed of a sigmoid neural net layer and a pointwise multiplication operation. The sigmoid layer outputs numbers between 0 and 1, describing how much of each component should be let through. A value of 0 means “let nothing through,” while a value of 1 means “let everything through!”An LSTM has three of these gates, to protect and control the cell state.

The first step of LSTM is to decide what information are to be thrown out from the cell state. It is made by a sigmoid layer called the “forget gate layer.” It looks at $h_{t-1}h_{t-1}$ and x_tx_t, and outputs a number between 00 and 11 for each number in the cell state $C_{t-1}C_{t-1}$. A 11 represents "completely keep this" while a 00 represents "completely remove this."

In the next step it is decided what new information are going to be stored in the cell state. It has two parts. First, a sigmoid layer called the "input gate layer" decides which values are to be updated. Thereafter, a *tanh* layer creates a vector of new candidate values, $C_{\sim t}C_{\sim t}$, that could be added to the state. In the next step, these two are combined to create an update to the state. It is now time to update the old cell state, $C_{t-1}C_{t-1}$, into the new cell state C_tC_t.

We multiply the old state by f_tf_t. Then we add $it{*}C_{\sim t}it{*}C_{\sim t}$. This is the new candidate values, scaled by how much we decide to update each state value.

Finally, we need to decide on the output. The output will be a filtered version of the cell state. First, we run a sigmoid layer which decides what parts of the cell state we're going to output. Then, we put the cell state through *tanh*-*tanh* (to push the values to be between −1−1 and 11) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

5 Proposed Framework to Forecast Share Price & Company Growth in Different Time Span

In this section, we shall first analyze some existing techniques and their merits to finally arrive at our methodology. Next, we shall discuss the algorithmic and implementation steps in detail. It is implemented in Python.

5.1 Analyzing Different Methods

Regression is one of the popular way to do the prediction of share prices. In figure 4 two figures on TCS share price using linear regression & polynomial regression of degree four are shown.

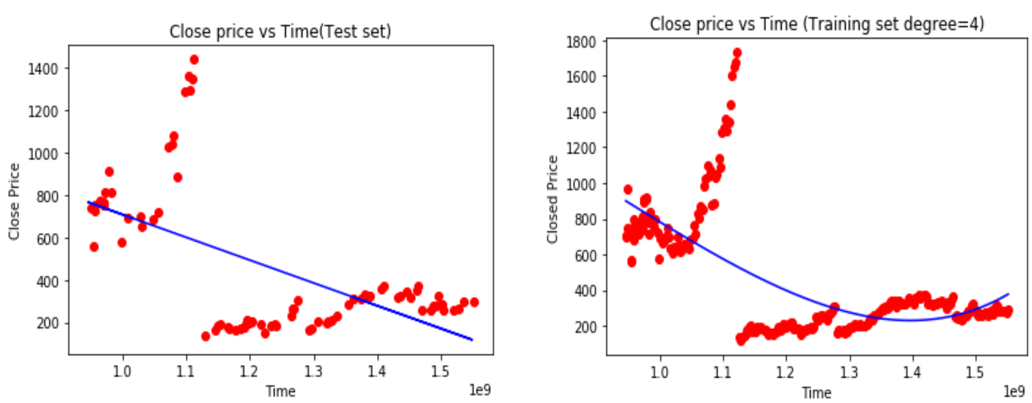

Figure 4: stock market closing prices of TCS over a time period and polynomial(degree 4) regression line

Regression is found not to be very much useful here to compute the error values. Also, we found a problem with curve fitting. The above graphs are showing a poor result in terms of curve fitting. This has a clear justification.For time series data, such as text, signals, stock prices, etc. LSTM is better suited to learn temporal patterns in deep neural networks. An LSTM solves the 'vanishing gradient' problem that exists in a RNN while learning long-term dependencies with time series dataset with the use of memory cell (states) and (input and forget) gates. So, LSTM may be a better option for future prediction of the company's share price as well as growth.

5.2 Methodology

The purpose of our framework is to analyze which is the best time span to predict the future share price of a company from a particular sector. Our objective is to predict the future price and calculate the future growth of the company in the different time span. Then we analyze the prediction error for each company of different sector. Based on that we conclude which time span is best for future prediction of that particular sector.

We first predict the future closing price of 5 different companies from some pre-decided sectors with the help of LSTM. This prediction will be done on historical data & the future prediction will be done for 3-month, 6-month, 1 year & 3 years. In these four different time spans (3 & 6 months, 1 & 3 years), we calculate the growth of those companies. Then by analyzing the deviations of closing price for each time span, we took the resultant time span which has maximum growth, i.e. less error for the particular sector, e.g. companies A, B, C, D & E from a sector S1 has more growth in 3-months' time span of prediction then we draw an conclusion that for sector S1, our framework gives the best prediction for next 3-months for that particular sector. In our analysis, let's consider we are using the data for *M*onths. Then the weight of a company is defined as:

```
weight = 1/ (P *(P+1)/2)
```

In our case, month-wise weight (Yi) will be calculated using the following algorithm:

```
N:= M
weight := 1/(M*(M+1)/2)
FOR i = 1 to M
Begin
Yi := weight * N ; /* Yi is the weight of previous ith month*/
Q = Q - 1;
i := i + 1
End
End FOR
```

Suppose the growth rate between different time periods is Gr_i where i=1 to M, considering current year as 0th year. Therefore, Gr_i is the growth rate of $(i\text{-}1)^{th}$ time period w.r.t its immediate earlier year i.e. i^{th} year. To maximize the impact of current growth over the growth of older year, we would develop a mathematical formula stated below. Suppose the growth rates of a company are Gr_1; $Gr_2 \ldots Gr_m$ respectively from present to M years earlier.

Then the Company Net Growth Rate (CNGR) by the following formula.

$$CNGR_j = Y_1 * Gr_1 + Y_2 * Gr_2 + \cdot\cdot\cdot + Y_i * Gr_i + \cdot\cdot\cdot + Y_p * Gr_m$$

Where $CNGR_j$ is the Company Net Growth Rate of the j^{th} company (where j=1 to m)

5.3 Implementation Steps

Step1: Raw Stock Price Dataset: Day-wise past stock prices of selected companies are collected from the BSE (Bombay Stock Exchange) official website.

Step*2: *Pre-processing: This step incorporates the following:

a) Data discretization: Part of data reduction but with particular importance, especially for numerical data

b) Data transformation: Normalization.

c) Data cleaning: Fill in missing values.
d) Data integration: Integration of data files. After the dataset is transformed into a clean dataset, the dataset is divided into training and testing sets so as to evaluate. Creating a data structure with 60 timesteps and 1 output.

Step3: Feature Selection: In this step, data attributes are chosen that are going to be fed to the neural network.In this study Date & Close Price are chosen as selected features.

Step 4: Train the NN model: The NN model is trained by feeding the training dataset. The model is initiated using random weights and biases. Proposed LSTM model consists of a sequential input layer followed by 3 LSTM layers and then a dense layer with activation. The output layer again consists of a dense layer with a linear activation function.

Step5: Output Generation: The RNN generated output is compared with the target values and error difference is calculated. The Backpropagation algorithm is used to minimize the error difference by adjusting the biases and weights of the neural network.

Step 6: Test Dataset Update: Step 2 is repeated for the test data set.

Step 7: Error and companies' net growth calculation: By calculating deviation we check the percentage of error of our prediction with respect to actual price.

Step 8**: **Visualization: Using Keras[21] and their function APIs the prediction is visualized.

Step 9: Investigate different time interval: We repeated this process to predict the price at different time intervals. For our case, we took 2-month dataset as training to predict 3-month, 6-month, 1 year & 3 years of close price of the share. In this different time span, we calculate the percentage of error in the future prediction. This would be different for different sectors. So, this will help to find a frame for the particular sector to predict future companies' net growth.

6 Results

The proposed LSTM based model is implemented using Python. In Table 1 the Error value for different companies belong to Banking Sector based on the historical data of 1 month, 3 month, 6 month, 1 Year, 3 Year span is shown.

Table 1: Error Value for Different Banks

Bank Names	1 month	3 month	6 month	1 year	3 year
SBI	93.30438	9.371283	19.5584	5.148866	0.830179
HDFC	532.8527	523.4962	162.8642	24.40721	0.987856
ICICI	71.80286	9.881709	10.76914	4.575525	0.863681
Avg Error	232.6533	180.9164	64.39726	11.3772	0.893905

Table 2: Error Value for Different Sectors

Sector	1 month	3 month	6 month	1 year	3 year
IT	39.56394	8.049353	1.48794	1.840666	0.782617
Pharma	250.7862	94.87654	29.48869	7.358529	0.903381
FMCG	426.7132	134.2102	60.45957	11.9643	0.874805
Aviation	291.025	35.08927	36.90103	30.97042	0.944595
Bank	232.6533	180.9164	64.39726	11.3772	0.893905

In the same way calculation is done for other sectors also based on the top level companies belong to that sector. The error values for the sector is shown in Table 2.

It has been observed from the result that for almost all the sectors the error level comes down drastically with the test data for longer periods. So we suggest to apply this LSTM based model to predict the share price on long time historical data.

7 Conclusions

In this paper, we analyze the growth of the companies from different sector and try to find out which is the best time span for predicting the future price of the share. So, this draws an important conclusion that companies from a certain sector have the same dependencies as well as the same growth rate. The prediction can be more accurate if the model will train with a greater number of data set.

Moreover, in the case of prediction of various shares, there may be some scope of specific business analysis. We can study the different pattern of the share price of different sectors and can analyze a graph with more different time span to fine tune the accuracy. This framework broadly helps in market analysis and prediction of growth of different companies in different time spans. Incorporating other parameters (e.g. investor sentiment, election outcome, geopolitical stability) that are not directly correlated with the closing price may improve the prediction accuracy.

References

[1] F. a. o. Eugene, "Efficient capital markets: a review of theory and empirical work," Journal of finance, vol. 25, no. 2, pp. 383-417, 1970.

[2] Z. A. Farhath, B. Arputhamary and L. Arockiam, "A Survey on ARIMA Forecasting Using Time Series Model," Int. J. Comput. Sci. Mobile Comput, vol. 5, pp. 104-109, 2016.

[3] S. Wichaidit and S. Kittitornkun, "Predicting SET50 stock prices using CARIMA (cross correlation ARIMA)," in 2015 International Computer Science and Engineering Conference (ICSEC), IEEE, 2015, pp. 1-4.

[4] D. Mondal, G. Maji, T. Goto, N. C. Debnath and S. Sen, "A Data Warehouse Based Modelling Technique for Stock Market Analysis," International Journal of Engineering & Technology, vol. 3, no. 13, pp. 165-170, 2018.

[5] G. Maji, S. Sen and A. Sarkar, "Share Market Sectoral Indices Movement Forecast with Lagged Correlation and Association Rule Mining," in International Conference on Computer Information

Systems and Industrial Management, Bialystok, Poland, Sprigner, 2017, pp. 327-340.

[6] M. Roondiwala, H. Patel and S. Varma, "Predicting stock prices using LSTM," International Journal of Science and Research (IJSR), vol. 6, no. 4, pp. 1754-1756, 2017.

[7] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data," PloS one, vol. 14, no. 2, p. e0212320, April 2019.

[8] S. Selvin, R. Vinayakumar, E. A. Gopalkrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in International Conference on Advances in Computing, Communications and Informatics, 2017.

[9] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen," Diploma, Technische Universität München, vol. 91, no. 1, 1991.

[10] Y. Bengio, P. Simard, P. Frasconi and others, "Learning long-term dependencies with gradient descent is difficult," IEEE transactions on neural networks, vol. 5, no. 2, pp. 157-166, 1994.

[11] S. Hochreiter and J. Schmidhuber, "LSTM can solve hard long time lag problems," in Advances in neural information processing systems, NIPS, 1997, pp. 473--479.

[12] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 6, no. 2, pp. 107-116, 1998.

[13] J. Schmidhuber, D. Wierstra, M. Gagliolo and F. Gomez, "Training recurrent networks by evolino," Neural computation, vol. 19, no. 3, pp. 757-779, 2007.

[14] L. Pasa and A. Sperduti, "Pre-training of recurrent neural networks via linear autoencoders," in Advances in Neural Information Processing Systems, NIPS, 2014, pp. 3572-3580.

[15] J. Chen and N. S. Chaudhari, "Segmented-memory recurrent neural networks," IEEE transactions on neural networks, vol. 20, no. 8, pp. 1267-1280, 2009.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[17] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, MIT Press, 2018.

[18] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, IEEE, 2000, pp. 189-194.

[19] K. Cho, B. Van Merri{\"e}nboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[20] K. Yao, T. Cohn, K. Vylomova, K. Duh and C. Dyer, "Depth-gated LSTM," arXiv preprint arXiv:1508.03790, 2015.

[21] J. Koutnik, K. Greff, F. Gomez and J. Schmidhuber, "A clockwork rnn," arXiv preprint arXiv:1402.3511, 2014.

[22] R. Kotikalapudi, "Keras Visualization Toolkit," [Online]. Available: https://raghakot.github.io/keras-vis. [Accessed 31 May 2019].